



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07302235 A**(43) Date of publication of application: **14 . 11 . 95**

(51) Int. Cl

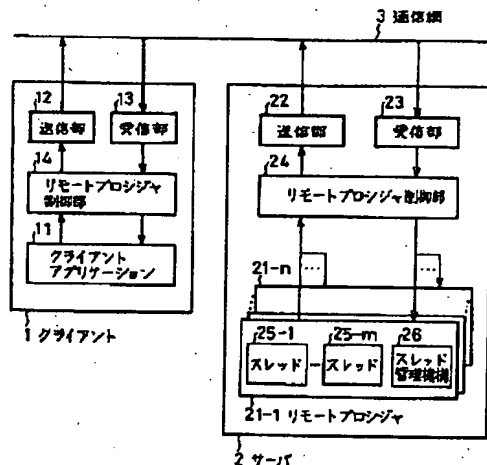
**G06F 13/00**(21) Application number: **06114602**(71) Applicant: **NEC CORP**(22) Date of filing: **30 . 04 . 94**(72) Inventor: **NISHIDE TOSHIHIRO**(54) **CLIENT-SERVER SYSTEM**

## (57) Abstract:

**PURPOSE:** To suppress the consumption of physical resources on a server side to minimum by turning a remote procedure on the server side to multi-thread constitution.

**CONSTITUTION:** When remote procedure calling for which an argument is specified is present from the client application 11 of a client 1, a remote procedure control part 14 transmits a request message including the argument and a remote procedure name to be called to a server 2 and the remote procedure control part 24 of the server 2 calls the pertinent remote procedure 21-1 among the plural remote procedures 21-1-21-n present on the side of the server 2. In the remote procedure 21-1, the thread management mechanism 26 newly generates a thread 25-1 corresponding to the calling and parallelly executes the plural threads including the other already generated threads.

COPYRIGHT: (C)1995,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-302235

(43) 公開日 平成7年(1995)11月14日

(51) Int.Cl.<sup>9</sup>

G 0 6 F 13/00

識別記号

3 5 7 Z 7368-5B

庁内整理番号

F I

技術表示箇所

審査請求 有 請求項の数 2 F D (全 5 頁)

(21) 出願番号 特願平6-114602

(22) 出願日 平成6年(1994)4月30日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 西出 俊浩

東京都港区芝五丁目7番1号 日本電気株式会社内

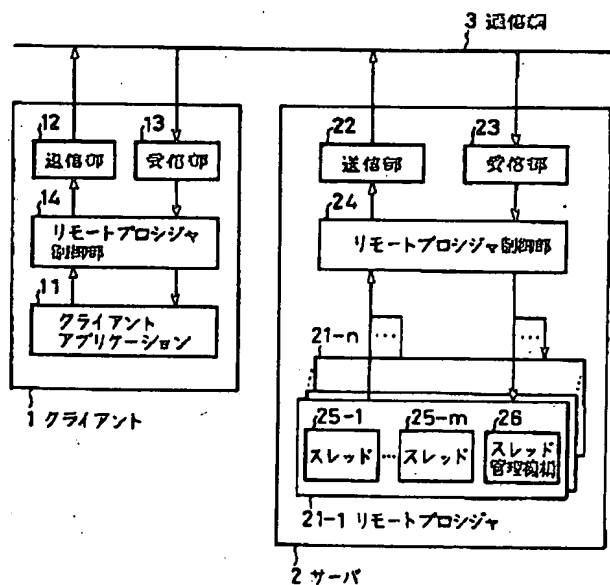
(74) 代理人 弁理士 境 廣巳

(54) 【発明の名称】 クライアント・サーバシステム

(57) 【要約】

【目的】 クライアント・サーバシステムにおいて、サーバ側のリモートプロシジャをマルチスレッド構成にすることにより、サーバ側の物理的リソースの消費を最小限に抑える。

【構成】 クライアント1のクライアントアプリケーション11から引数を指定したリモートプロシジャ呼び出しがあると、リモートプロシジャ制御部14は引数および呼び出すべきリモートプロシジャ名を含む要求メッセージをサーバ2に送信する。サーバ2のリモートプロシジャ制御部24は、サーバ2側に存在する複数のリモートプロシジャ21-1~21-nのうちの該当するリモートプロシジャ21-1を呼び出す。リモートプロシジャ21-1では、そのスレッド管理機構26が、その呼び出しに対応する1つのスレッド25-1を新たに生成し、既に生成済の他のスレッドを含め複数のスレッドを並行して実行する。



## 【特許請求の範囲】

【請求項1】 処理を要求するクライアントと処理を要求されるサーバとが通信網に接続されたクライアント・サーバシステムにおいて、

前記クライアントに、

前記サーバとの間で前記通信網を介して通信するためのクライアント側送受信部と、

処理を要求するクライアントアプリケーションと、

該クライアントアプリケーションからの引数を指定したリモートプロシジャ呼び出しにตอบสนองして前記クライアント側送受信部および前記通信網を通じて前記サーバに対し前記引数および呼び出すべきリモートプロシジャ名を含む要求メッセージを送信し、また前記クライアント側送受信部および前記通信網を通じて前記サーバから受信した応答メッセージ中の処理結果を前記クライアントアプリケーションに通知するクライアント側リモートプロシジャ制御部とを備え、

前記サーバに、

前記クライアントとの間で前記通信網を介して通信するためのサーバ側送受信部と、

おのおのがマルチスレッド構造の複数のリモートプロシジャと、

前記サーバ側送受信部で受信された前記要求メッセージにตอบสนองして前記複数のリモートプロシジャのうちの該当するリモートプロシジャを呼び出し、またリモートプロシジャの処理結果を含む応答メッセージを生成して前記サーバ側送受信部および前記通信網を通じて前記クライアントに送信するサーバ側リモートプロシジャ制御部とを備えることを特徴とするクライアント・サーバシステム。

【請求項2】 前記サーバ側の個々のリモートプロシジャ中のスレッドを管理するスレッド管理機構を備え、前記スレッド管理機構は、リモートプロシジャの呼び出しを受けた時点でその呼び出しに対応する1つのスレッドを新たに生成し、既に生成済の他のスレッドを含め複数のスレッドを並行して実行することを特徴とする請求項1記載のクライアント・サーバシステム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、処理を要求するクライアントと処理を要求されるサーバとが通信網に接続されたクライアント・サーバシステムに関する。

## 【0002】

【従来の技術】 従来のこの種のクライアント・サーバシステムとしては、図3に示すようなものが知られている。同図において、100は処理を要求するクライアント、200は処理を要求されるサーバであり、両者は通信網300によって相互に接続されている。クライアント100には、実際の処理要求を行うクライアントアプリケーション101、メッセージ組立部102、メッセ

ージ解析部103、送信部104および受信部105が備えられ、サーバ200には、クライアント100からの処理要求を実際に処理する複数のサーバアプリケーション201-1~201-n、メッセージ振分部202、メッセージ解析部203、メッセージ組立部204、送信部205および受信部206が備えられている。

【0003】 クライアントアプリケーション101は、何れかのサーバアプリケーション、例えばサーバアプリケーション201-1に対して処理を要求する場合、自クライアント100上に存在するアプリケーションに対して処理を要求する際の呼び出し手法とは異なり、要求する処理の内容をメッセージ組立部102に通知する形式で処理を要求する。メッセージ組立部102は、要求された処理内容をメッセージとして組み立て、送信部104から通信網300を通じてサーバ200に送信する。この処理要求のメッセージはサーバ200の受信部206で受信されてメッセージ解析部203に送られ、このメッセージ解析部203でどのような処理内容であるかが解析され、この解析結果に基づきメッセージ振分部202が該当するサーバアプリケーション201-1に処理を要求する。処理を要求されたサーバアプリケーション201-1での処理が終了すると、メッセージ組立部204はその処理結果をメッセージとして組み立て、送信部205および通信網300を通じてクライアント100に送信する。この処理結果のメッセージはクライアント100の受信部105で受信され、メッセージ解析部103によって解析されて、要求元のクライアントアプリケーション101に処理結果が通知される。

【0004】 上述した従来のクライアント・サーバシステムでは、クライアントアプリケーション101は、自クライアント100上に存在するアプリケーションに対して処理を要求する際の呼び出し手法とは別の形式でサーバ200に対して処理を要求する必要があり、クライアントアプリケーション101の設計が面倒になるという問題点がある。

【0005】 そこで、このような問題点を解消するために、クライアント・サーバシステムにおける要求、応答型の通信を通常の手続き呼び出しと同じように行えるようにしたリモートプロシジャ呼び出し(RPC)が提案されている。このリモートプロシジャ呼び出しでは、通常のプロシジャ呼び出しと同じようにリモートプロシジャを呼び出せるため、クライアントアプリケーションの設計が容易になる。なお、呼び出したプログラムの実行は、呼び出されたプロシジャの実行が終了するまで中断される。また、呼び出しに際してパラメータを指定することができ、プロシジャは値を返すことができる。

## 【0006】

【発明が解決しようとする課題】 ところで、リモートプ

ロシジャ呼び出しを適用したクライアント・サーバシステムにおいて、システムのスループットを高めるには、複数のクライアントアプリケーションが同時期に同種の呼び出しを行った場合、それらの呼び出しにかかるリモートプロシジャの処理を並行して処理し得るようにする必要がある。しかし、これを実現するために、同種のリモートプロシジャをサーバ上に複数備える構成によれば、各々のリモートプロシジャはそのコード部やデータ部等を格納するために別個のメモリ等の物理的リソースを必要とするため、サーバの物理的リソースが圧迫されるという問題点がある。

【0007】本発明はこのような事情に鑑みて提案されたものであり、その目的は、サーバ側の物理的リソースの消費を最小限に抑えることができるクライアント・サーバシステムを提供することにある。

【0008】

【課題を解決するための手段】本発明は上記の目的を達成するために、サーバ側の各々のリモートプロシジャをマルチスレッド構造のものとしている。そして、リモートプロシジャの呼び出しを受けた時点でその呼び出しに対応する1つのスレッドを新たに生成し、既に生成済の他のスレッドを含め複数のスレッドを並行して実行するように構成している。

【0009】

【作用】本発明のクライアント・サーバシステムにおいては、クライアントのクライアントアプリケーションから引数を指定したリモートプロシジャ呼び出しがあると、クライアント側リモートプロシジャ制御部が、クライアント側送受信部および通信網を通じてサーバに対し前記引数および呼び出すべきリモートプロシジャ名を含む要求メッセージを送信し、この要求メッセージを受信したサーバ側リモートプロシジャ部が、サーバ側に存在する複数のリモートプロシジャのうちの該当するリモートプロシジャを呼び出す。呼び出されたりリモートプロシジャでは、そのスレッド管理機構が、その呼び出しに対応する1つのスレッドを新たに生成し、既に生成済の他のスレッドを含め複数のスレッドを並行して実行する。そして、或るスレッドの実行により処理結果が得られると、サーバ側リモートプロシジャ制御部がその処理結果を含む応答メッセージを生成してサーバ側送受信部および通信網を通じてクライアントに送信し、この応答メッセージを受信したクライアント側リモートプロシジャ制御部が、受信した応答メッセージ中の処理結果をクライアントアプリケーションに通知する。

【0010】

【実施例】次に本発明の実施例について図面を参照して詳細に説明する。

【0011】図1は本発明のクライアント・サーバシステムの一実施例のブロック図である。同図において、1は処理を要求するクライアント、2は処理を依頼される

サーバであり、クライアント1とサーバ2とは通信網3によって相互に接続されている。クライアント1には、リモートプロシジャ呼び出しによって実際に処理要求を行うクライアントアプリケーション11、サーバ2との間で通信網3を通じて通信するための送信部12および受信部13、クライアントアプリケーション11から出されたりリモートプロシジャ呼び出しを処理するリモートプロシジャ制御部14が設けられ、サーバ2には、クライアント1からの処理要求を実際に処理する複数のリモートプロシジャ21-1~21-n、クライアント1との間で通信網3を通じて通信するための送信部22および受信部23、クライアント1から出されたりリモートプロシジャ呼び出しを処理するリモートプロシジャ制御部24が設けられている。

【0012】リモートプロシジャ21-1~21-nは、それぞれ異なる処理を実行するプロシジャであり、複数のクライアントアプリケーションからの呼び出しを並行して処理し得るように、リモートプロシジャ21-1中に例示するスレッド25-1~25-mの如く、それぞれスレッドが複数存在し得るマルチスレッド構成を有している。ここで、スレッドとは、計算の実体であり、サーバ2を構成するプロセッサの状態（レジスタ、プログラムカウンタ、スタックポインタ、特権レベル等）を含むコンテキストで構成される。同一リモートプロシジャ内の複数のスレッドは、コード部やデータ等を他のスレッドと共有するため、少ないメモリ資源で複数の処理を多重に実行することができる。スケジューリングは、スレッド管理機構26によってスレッドを単位として行われる。

【0013】図2にスレッド管理機構26の処理例を示す。スレッド管理機構26は、クライアントアプリケーション11からのリモートプロシジャへの接続要求時にリモートプロシジャ内にクライアントアプリケーション11に対応してスレッドを生成する（S1）。スレッド管理機構26は、生成したスレッドを含めて次に実行すべきスレッドを決定し（S2）、次いで、必要に応じて現在実行中のスレッドの実行環境を退避する等のスレッド実行環境の入れ替えを行って、これから実行すべきスレッドの実行環境をリモートプロシジャ内に生成する（S3）。実行環境が整いスレッドが実行可能な状態になると、スレッド管理機構26はスレッドを実行する（S4）。そして、スレッドがプロセッサに依存しない命令である、例えばディスクI/O等のI/O要求を行ったとき、或いは予め定められたプロセッサ時間を消費したタイムアウト時（S5、S6）は、処理S2に戻り、次に実行すべき優先度の高いスレッドに処理を移す。また、或るスレッドの実行中に、別のリモートプロシジャ呼び出しにより他のスレッドの生成要求が生じた場合（S7）、現在のスレッドの実行を中断させてスレッドの生成処理S1を実行し、処理S2に戻って次に実

行すべきスレッドを決定する。

【0014】以下、上述のように構成された本実施例のクライアント・サーバシステムの動作を説明する。

【0015】クライアント1のクライアントアプリケーション11から、例えばリモートプロシジャ21-1のプロシジャ名および引数を指定したリモートプロシジャ呼び出しが発生すると、制御がリモートプロシジャ制御部14に移り、クライアントアプリケーション11はその時点で中断せしめられる。

【0016】リモートプロシジャ制御部14は、クライアントアプリケーション11から出された引数を送信し、  
10 やすい形にメッセージ化し、それを、呼び出すリモートプロシジャ21-1のプロシジャ名と呼び出し元のクライアントアプリケーション11の名前と共に、要求メッセージとして送信部12および通信網3を通じてサーバ2に送信する。

【0017】サーバ2の受信部23は、メッセージ化された引数と呼び出すリモートプロシジャ21-1のプロシジャ名と呼び出し元のクライアントアプリケーション11の名前とを含む要求メッセージを受信すると、これ  
20 をリモートプロシジャ制御部24に通知し、リモートプロシジャ制御部24は、メッセージ化された引数を元の状態に展開し、呼び出し対象となるリモートプロシジャ21-1に対し、呼び出し元のクライアントアプリケーション11の名前と共にそれらの引数を引き渡す。

【0018】リモートプロシジャ21-1のスレッド管理機構26は、呼び出し元のクライアントアプリケーション11の名前と引数とが渡されると、図2に示す処理  
30 S1で、当該クライアントアプリケーション11の名前に対応するスレッドを生成する。そして、今回生成したスレッドを含めて次に実行すべきスレッドを決定し(S2)、次に、実行すべきスレッドの実行環境をリモートプロシジャ21-1内に生成し(S3)、スレッドを実行する(S4)。そして、この実行中のスレッドがI/O要求を行ったとき、或いは予め定められたプロセッサ時間を消費したとき(S5、S6)、処理S2に戻って次に実行すべき優先度の高いスレッドに処理を移す。この結果、リモートプロシジャ21-1内では、このリモートプロシジャ21-1を呼び出したクライアントアプリケーションの数に等しいスレッドが生成され、リモート  
40 プロシジャ21-1内で時分割的に順次処理されることになる。

【0019】さて、リモートプロシジャ21-1の或るスレッド、例えばスレッド25-1で処理が終了する

と、その処理結果と要求元のクライアントアプリケーション11の名前とがリモートプロシジャ制御部24に通知される。リモートプロシジャ制御部24は、処理結果を送信に適するようにメッセージ化し、クライアントアプリケーション11の名前と共に、応答メッセージとして送信部22および通信網3を通じてクライアント1に送信する。

【0020】この応答メッセージはクライアント1の受信部13で受信されてリモートプロシジャ制御部14に通知され、リモートプロシジャ制御部14は、呼び出し  
10 元のクライアントアプリケーション11を再開せしめて、メッセージ化された処理結果をリモートプロシジャ呼び出しのリモート値としてクライアントアプリケーション11に返却する。

【0021】

【発明の効果】以上説明したように、本発明は、サーバ側のリモートプロシジャをマルチスレッド構造にして、リモートプロシジャ内の複数のスレッドによって各クライアントアプリケーションの処理を各スレッドで分担して処理するようにしたので、スループットを低下させることなくサーバ側の物理的リソースの消費を最小限に抑えることができる効果がある。

【図面の簡単な説明】

【図1】本発明のクライアント・サーバシステムの一実施例のブロック図である。

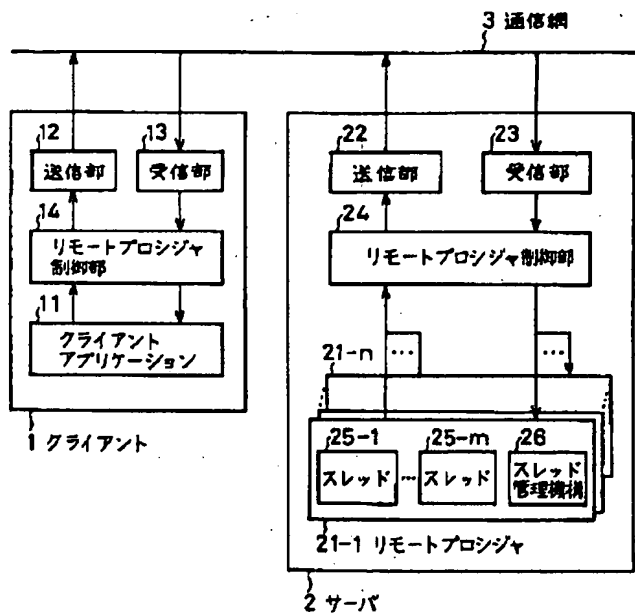
【図2】スレッド管理機構の処理例を示すフローチャートである。

【図3】従来のクライアント・サーバシステムの構成例を示すブロック図である。

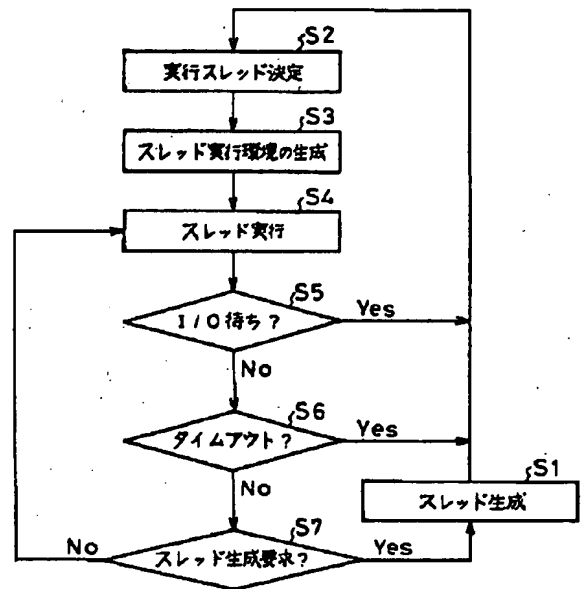
【符号の説明】

- 1…クライアント
- 11…クライアントアプリケーション
- 12…送信部
- 13…受信部
- 14…リモートプロシジャ制御部
- 2…サーバ
- 21-1～21-n…リモートプロシジャ
- 22…送信部
- 23…受信部
- 24…リモートプロシジャ制御部
- 25-1～25-m…スレッド
- 26…スレッド管理機構
- 3…通信網

【図 1】



【図 2】



【図 3】

